Agent Aware Organizational Design

JASON SLEIGHT, University of Michigan

Categories and Subject Descriptors: I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Coherence & Co-ordination, Multiagent Systems

General Terms: algorithms, performance, design, theory

Additional Key Words and Phrases: organizational design; Dec-MDP

1. INTRODUCTION

As multiagent systems (MASs) such as distributed sensor nets or multi-robot systems increase in size, interconnectivity, complexity, and longevity, coordinating the agents’ reasoning and behaviors becomes increasingly difficult. One approach to address these issues is to use insights from human organizations to design structures within which the agents can more efficiently reason and interact (see Dignum and Padget [2012] for a recent overview of such research). Generally speaking, an organization provides influences to each agent such that, by following its influences, an agent can make globally-useful local decisions without having to explicitly reason about the complete joint coordination problem. For example, an organizational influence might constrain and/or inform which actions an agent performs. If these influences are constructed to be cohesive and correlated across the agents, then each agent is influenced into performing only the actions that are appropriate for its (organizationally-designated) portion of the joint problem.

While prior work has largely focused on developing modeling languages to allow expert humans to specify an organizational representation for a MAS [Dignum et al. 2005; Hübner et al. 2007; Horling and Lesser 2008], or on adaptive processes to allow (typically implicit) organizations to emerge via repeated interactions with the environment [Bowling and Veloso 2002; Dorigo et al. 2006], I instead investigate how an automated organizational design process (ODP) can create computational organizations. By leveraging details about how the agents reason about their decisions I establish quantitative, principled metrics for organizational performance based on the expected impact that an organization will have on the agents reasoning and behaviors. I define the organizational design space as the possible ways in which an organization could impact the agents’ reasoning and behaviors, and use my performance metrics to characterize the organizational design problem as search of this design space. Unsurprisingly, my analysis reveals that creating a provably optimal organization is computationally intractable, and thus I develop techniques that improve the efficiency of my ODP via approximating the incremental impact of a proposed organizational influence. My empirical results demonstrate that my ODP can create organizations that not only steer an agent into behaving in ways that mesh well with other agents’ behaviors, but also can simplify an agent’s reasoning about such behaviors. Finally, I look at how an ODP can inform agents’ organizational adaptation decisions (e.g., in the event there is a mismatch between the ODP’s expectations and the actual execution environment) by providing second-order information about its organizational design.

2. ORGANIZATIONAL DESIGN PROCESS

In contrast to prior, problem-centric approaches where organizational design is viewed as decomposing and solving a problem as a MAS [Dignum and Padget 2012], my agent-centric approach views organizational design as influencing agents’ reasoning and behaviors to impart desirable coordination patterns upon a MAS. The idea is that, by grounding decisions in the agents’ reasoning and behaviors, an ODP can reason about the expected performance of candidate organizations in a principled, quantitative fashion that directly mirrors how the
organization is expected to impact the MAS in the execution environment. My methodology for creating an organizational specification language follows from this philosophy. First, I commit to a specific agent reasoning framework, in this case I have elected to use the decentralized Markov decision process (Dec-MDP) due to its generality for a wide range of problem domains, and its principled, well-understood mathematical formalism [Bernstein et al. 2002]. Then, I use the formalism of the decided-upon reasoning framework to enumerate the ways in which an organizational design could possibly affect the agents’ reasoning and/or behaviors, which defines my organizational specification language. For the case of Dec-MDPs, the specification language consists of modifications to the agents’ state representations, action spaces, transition/reward functions, etc. [Sleight and Durfee 2012; 2013]. An organizational influence \( \Delta \) in my organizational specification language, could for example, prohibit an agent \( i \) from considering an action \( a_i \) in a state \( s \), which prevents the agent from considering any policy \( \pi : S \rightarrow A_i \), where \( \pi(s) = a_i \).

To search through the space of organizational designs (as defined by the organizational specification language), an ODP needs a way to measure the quality of candidate organizations. Beyond the traditional behavior quality metric (e.g., reward in a decision theoretic framework) emphasized by other research, my agent-grounded organizational formulation reveals that the computational requirements imparted on the agents by an organization are also a valuable metric of organizational performance. Since my specification language is directly derived from the agents’ reasoning framework, by definition an organizational influence has principled impact on both the agents’ behavior quality and computational costs, and can thus be directly measured by an ODP [Sleight and Durfee 2014]. An important point here is that my metrics quantify the expected performance of the organization on the collective system performance, and thus enable the ODP to formulate an organization consisting of cohesive, jointly correlated organizational influences. Returning to the example organizational influence \( \Delta \) that blocks \( a_i \) from consideration in \( s \), \( \Delta \) could reduce the size of agent \( i \)’s reachable state space thus reducing its local computational costs (in a well-defined manner), as well as increase the expected joint reward if agent \( j \neq i \) should be responsible for that behavior instead of agent \( i \) (again in a well-defined manner).

My ODP algorithm [Sleight and Durfee 2013; 2014] begins by utilizing a global perspective of the domain model and solves for optimal joint policies in a sub-sampling of the joint problem space, which serves to inform the ODP of what constitutes a good collective behavior. Then, the ODP estimates the incremental performance impact of an organizational influence by computing summary statistics of the expected joint reward and computational costs associated with that influence over the sampled problem instances. The ODP uses this data to search through the space of organizational designs. Since there is a distinct organizational design for every way in which the agents’ reasoning and behaviors can be organizationally influenced (size is combinatorial in the number of joint policies), it is unsurprising that finding a provably optimal organization is computationally intractable [Sleight and Durfee 2014]. For this reason, my ODP algorithm utilizes a greedy, incremental search as opposed to an exhaustive one, and finds an approximately optimal organization that imparts long-term, desirable patterns of reasoning and behaviors on the MAS.

Since my ODP approach relies on utilizing a global perspective of the domain model, one risk is that if the ODP’s global model is inaccurate (e.g., the agents possess individual expertise or the ODP is simply unaware of the complete, detailed global problem), then the ODP’s design could be poorly suited for the actual execution environment. I mitigate this concern in two primary ways: by providing abstract influences rather than fully detailed ones, and by supporting organizational adeptness. Prior research [Dignum et al. 2005] suggests that providing abstract influences as opposed to detailed micromanagement can yield more robust organizations. For my ODP approach, I have shown that focusing on influences to the possible inter-agent dependencies biases the ODP away from micromanaging the
agents and can yield more robust organizations in the face of inaccuracies within the ODP’s model [Sleight and Durfee 2013].

The second method I investigate to mitigate the effect of ODP-model inaccuracies is organizationally adept agents [Corkill et al. 2011], agents who use second-order information about their organization to intelligently reason about adaptations to their current organization. The focus of my research, in this aspect, is on how an ODP can supply the necessary supplementary information alongside an organization to enable such agent adeptness. My approach centers on the observation that an organizational design is conditional on certain expectations, both in that an organization is created based on certain environmental expectations (e.g., domain dynamics, agent capabilities, etc.), and that an organization imparts expectations on the MAS (e.g., task responsibilities, interaction patterns, etc.). So long as the ODP’s expectations are met, then the organization it creates is appropriate for the agents and should be followed, and only when the environment deviates from those expectations should the agents make adaptations to their organization [Sleight and Durfee 2012]. Thus, the supplementary information that an ODP provides should inform the agents of the underlying expectations of the organizational design.

3. FUTURE WORK

While my preliminary investigations have thus far demonstrated the promise of my approach, there are several remaining challenges to overcome. Although it is intuitively reasonable that my organizational specification language defines a principled organizational design space that stems from the agents’ reasoning framework, it is not obvious that all of the language constructs are necessary, or if other constructs should be added. Given that I have committed to a specific agent reasoning framework, however, it is possible to analytically prove the necessity and completeness properties of my language (w.r.t. the reasoning framework), which would conclusively define the space of influences my ODP can consider (for my specific agent reasoning framework).

The theoretical worst-case complexity of my organizational design problem is exceptionally high, $O(|\pi|!)$ for a joint policy space with cardinality $|\pi|$, meaning that efficient approximation algorithms are important for practical application of my methods. To date, I have identified several general-purpose properties than can simplify organizational design search. For example, if an ODP can compute the incremental impact of an individual organizational influence (rather than wholesale evaluation of a candidate organization’s performance), then it can embed those calculations within incremental search algorithms (e.g., greedy hill climbing, Monte Carlo, $A^*$, etc.). While the incremental impact of an organizational influence is well-defined by the agents’ reasoning framework, efficiently computing this information is non-trivial since the incremental impact of an influence is conditional on the current candidate organization. My research has revealed how to make these incremental calculations independent of the current candidate organization for action influences (which allows an ODP to efficiently search through the action influence space), but more work remains to extend my ODP to efficiently search through the remaining influence forms.

In principle, an ODP could provide all of its expectations alongside an organizational design, and let the agents figure out which of the expectations are important and which have insignificant consequence. This could be undesirable, however, if the number of expectations is large, as is likely to be the case since the ODP has an expectation associated with each transition (of which there are $O(S_i \times A_i \times S_i \times n)$), each reward (of which there are $O(S_i \times A_i \times S_i \times n)$), etc. Thus, rather than simply providing the agents with every organizational expectation, an ODP should limit the second-order information it provides to only that which is useful for the agents in determining how they should adapt their organization. Characterizing this subset of expectations is another important aspect for future work.
REFERENCES


